EXHIBIT 1

Paper No. 1

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

KAYAK SOFTWARE CORP., OPENTABLE, INC.,

PRICELINE.COM LLC, and THE PRICELINE GROUP INC.,

Petitioner

v.

INTERNATIONAL BUSINESS MACHINES CORP.

Patent Owner

Inter Partes Review No. IPR2016-00604

Patent 5,961,601

IPR No. IPR2016-00604

Patent 5,961,601

Petitioner's Exhibit List

Exhibit	Description
1001	U.S. Patent No. 5,961,601 to Iyengar
1002	Declaration of Ray R. Larson
1003	United States Patent No. 5,717,860 to Graber
1004	Bertrand Ibrahim, "World-Wide Algorithm Animation," Computer Networks and ISDN Systems Volume 27, pp. 255-265 (1994) ("Ibrahim")
1005	U.S. Patent No. 6,016,484 to Williams ("Williams")
1006	Louis Perrochon et al., "IDLE: Unified W3-access to interactive information servers"
1007	"The World Wide Web and Emerging Internet Resource Discovery Standards for Scholarly Literature," Stuart L. Weibel, Library Trends Volume 43, No. 4, pp. 627-644 (Spring 1995) ("Weibel")
1008	Excerpts of Ian S. Graham, The HTML Sourcebook (1995)
1009	Tim Berners-Lee et al., World-Wide Web: The Information Universe, Electronic Networking, Vol. 2, no. 1, at 56 (Spring 1992)
1010	Prosecution of '601 Patent: March 25 1998 Office Action
1011	Prosecution of '601 Patent: July 6, 1998 Response to Office Action
1012	Prosecution of '601 Patent: September 16, 1998 Office Action

1013	Prosecution of ’601 Patent: January 11, 1999 Response to Office Action
1014	Prosecution of ’601 Patent: Notice of Allowability for U.S. Patent No. 5,961,601
1015	Excerpts of IBM’s Opposition to Defendants’ Motion to Dismiss
1016	Prosecution of ’601 Patent: Information Disclosure Statement

TABLE OF CONTENTS

I. INTRODUCTION 1

II. MANDATORY NOTICES 2

A. Real Party in Interest (37 C.F.R. § 42.8(b)(1)) 2

B. Related Matters (37 C.F.R. § 42.8(b)(2)) 2

C. Lead and Back-Up Counsel (37 C.F.R. § 42.8(b)(3)) 2

D. Service Information (37 C.F.R. § 42.8(b)(4)) 3

III. GROUNDS FOR STANDING 3

IV. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED 3

A. Claims for which Review is Requested (37 C.F.R. § 42.104(b)(1)) 3

B. Statutory Grounds of Challenge (37 C.F.R. § 42.104(b)(2)) 4

V. REASONS FOR THE RELIEF REQUESTED UNDER 37 C.F.R. §§ 42.22(A)(2) AND 42.104(B)(4) 5

A. Background 5

B. Claim Construction (37 C.F.R. § 42.104(b)(3)) 9

C. Level of Ordinary Skill in the Art 13

D. The Challenged Claims Are Invalid Under §§ 102(a) or 103(a) 14

1. Ground 1 – Claims 1, 2, 6, 7, 8, 11, 12, 14, 15, 20, 21, 25, 27, 28, 33, 34, 38, 40, 41, 47, 49, 51, 52, 57, 60, 61, and 66 are rendered obvious by Graber 14

2. Ground 2: Graber view of Ibrahim renders claims 3, 16, 29, 42, 53, and 62 obvious 49

3. Ground 3: Graber in view of Ibrahim and Williams renders claims 4, 5, 17, 18, 30, 31, 43, 44, 63, and 64 obvious 53

VI. CONCLUSION 60

I. INTRODUCTION

Pursuant to 35 U.S.C. § 312 and 37 C.F.R. § 42.100 *et seq.*, Kayak Software Corp., OpenTable, Inc., Priceline.com, LLC, and The Priceline Group Inc. (collectively, "Petitioner") request *Inter Partes* Review of claims 1-8, 11, 12, 14-21, 24, 25, 27-34, 37, 38, 40-45, 47-49, 51-57, and 60-66 (the "Challenged Claims") of U.S. Patent No. 5,961,601 ("the '601 Patent," Ex. 1001). The Board is authorized to deduct all required fees for this Petition from Norton Rose Fulbright Deposit Account No. 06-2375.

The claims of the '601 Patent describes a technique for computers maintaining state information while communicating over the World Wide Web ("WWW"). The protocol used on the WWW, HTTP, is stateless, meaning every communication is treated independently; but certain WWW applications benefit from having access to information about past communications. The '601 Patent admits that both the problem and the specific solution it proposes—embedding state in communications between client and server by including the state in a dynamically generated HTML document provided to a client—were already known in the prior art. At best, the '601 Patent purports to claim a specific method

of server-side embedding state information in a hyperlink. But, as further described below, the implementation details in the Challenged Claims were well-known.

The Petition and accompanying evidence demonstrate that the challenged claims are unpatentable. Accordingly, Petitioner respectfully requests that the Board institute trial on the grounds set forth herein.

II. MANDATORY NOTICES

A. Real Party in Interest (37 C.F.R. § 42.8(b)(1))

KAYAK Software Corp., OpenTable, Inc., Priceline.com LLC, and The Priceline Group Inc. are the real parties-in-interest.

B. Related Matters (37 C.F.R. § 42.8(b)(2))

The following matter may affect, or be affected by, a decision in this proceeding: *International Business Machines Corporation v. The Priceline Group Inc. et al.*, Civil Action No. 1:15-cv-137 (D.E.D. Feb. 9, 2015) (the "Litigation").

C. Lead and Back-Up Counsel (37 C.F.R. § 42.8(b)(3))

Lead counsel: Richard S. Zembek (Reg. No. 43,306)

Back-up counsel: Gilbert A. Greene (Reg. No. 48,366)

D. Service Information (37 C.F.R. § 42.8(b)(4))

Please address all correspondence and service to: Richard Zembek, Norton

Rose Fulbright US LLP, 1301 McKinney, Suite 5100, Houston, TX 77010, Ph: 713.651.5283, Fax: 713.651.5246.

Petitioner consents to electronic service by email at:

Priceline-IBM-NRFService@nortonrosefulbright.com.

III. GROUNDS FOR STANDING

Pursuant to 37 C.F.R. § 42.104(a), Petitioner certifies that the '601 Patent is available for *Inter Partes* Review, and that Petitioner is not barred or estopped from requesting an *Inter Partes* Review challenging the Challenged Claims on the grounds identified in this Petition. The '601 Patent has not been subject to a previous estoppel-based proceeding of the AIA, and Petitioner was served with the original complaint in the above-referenced Litigation within the last 12 months.

IV. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED

A. Claims for which Review is Requested (37 C.F.R. § 42.104(b)(1))

Petitioner requests the review and cancellation as invalid of claims 1-8, 11, 12, 14-21, 24, 25, 27-34, 37, 38, 40-45, 47-49, 51-57, and 60-66 of the '601 patent (the "Challenged Claims").

B. Statutory Grounds of Challenge (37 C.F.R. § 42.104(b)(2))

For the reasons presented below, Petitioner seeks the following relief:

Ground 1: Invalidation of Claims 1, 2, 6-8, 11, 12, 14, 15, 20, 21, 25, 27, 28, 32, 33, 34, 37, 38, 40, 41, 45, 47-49, 51, 52, 56, 57, 60, 61, 65, and 66 under 35 U.S.C. § 103(a) by United States Patent No. 5,717,860 to Graber. Graber was filed on September 20, 1995, and issued February 10, 1998, making it prior art to the '601 Patent (which was filed June 7, 1996) under at least § 102(e).

Ground 2: Invalidation of Claims 3, 16, 29, 42, 53, and 62 under 35 U.S.C. § 103(a) by Graber in view of "World-Wide Algorithm Animation" by Bertrand Ibrahim ("Ibrahim"). Ibrahim was published in 1994 in Computer Networks and ISDN Systems Volume 27, making it prior art under at least 35 U.S.C. § 102(b). The publication date is indicated in the heading of the article. Moreover, Patent Owner submitted Ibrahim as prior art during prosecution. *See* Ex. 1016.

Ground 3: Invalidation of Claims 4, 5, 17, 18, 30, 31, 43, 44, 54, 55, 63, and 64 under 35 U.S.C. § 103(a) based on Graber in view Ibrahim and U.S. Patent No. 6,016,484 to Williams. Williams was filed on April 26, 1996, and issued on January 18, 2000, rendering it prior art to the '601 Patent under at least § 102(e).

V. REASONS FOR THE RELIEF REQUESTED UNDER 37 C.F.R. §§ 42.22(A)(2) AND 42.104(B)(4)

A. Background

1. The '601 Patent admits the purported problem and

proposed solution were both known

The ’601 Patent generally relates to “preserving state in computers communicating over networks, such as the WWW using stateless protocols, e.g., HTTP.” Ex. 1001 at Abstract. “Statelessness” means a server receiving a request does not remember previous requests from a user. Ex. 1002 at ¶ 23. Within such networks, such as the WWW, a client will have a sequence of communications with “one or more servers,” and that “hypertext links to documents on both local and remote servers can be placed in an HTML file.” Ex. 1001 at 9:52; 6:51-53. Accordingly, the patent recites a need to “preserve state information in a conversation between a client adapted to request services from *one or more servers*” Ex. 1001 at 9:52-62 (emphasis added).

The ’601 Patent acknowledges that preserving state in the stateless environment of HTTP such that a client-server exchange can build upon previous exchanges was known. Ex. 1001 at 7:33-9:37. In fact, the patent’s background section titled “Current Methods for Handling State on the Web.” *Id*. Two admitted prior art “Methods for Handling State on the Web” are particularly relevant. First, in the admitted prior art “Web servers typically preserve state by passing state variables as hidden variables within forms.” *Id.* at 8:5-8. The “web server”

executes a CGI program that dynamically generates a web page with the state information embedded. *Id.* at 8:9-19 ("The form is generated dynamically and contains the user and session ID's as hidden variables."); *id.* at 4:36-40 ("CGI programming is well known in the art."). The '601 Patent criticizes this method for not permitting use of state without a "Form." *Id.* at 8:33-46.

Second, the '601 Patent admits that it was also known that state information could be passed without using the "Form" feature "by passing arguments" in a URL. *Id.* at 7:50-55 ("the command, http://trnaman.watson.ibm.com/cgibin/getargs?var1=7 & var2=10 invokes a CGI program passing the variables var1=7 and var2=10"). The patent states that a problem with this approach is that the client must follow the exact syntax for passing variables to CGI programs. *Id.* at 7:55-57. This criticism of the prior art incorrectly suggests that it was only known for a *client* to dynamically create such a URL with embedded state information. Ex. 1002 at ¶ 26.

In fact, several sources describe this approach—dynamically generating web pages with state information embedded in each URL to solve the problem of maintaining state in stateless HTTP. Graber teaches a method whereby state information (specifically the identity of a referring website) is appended "to the

end of the URL" passed back to the user. Ex. 1003 at 12:44-48. Similarly, Ibrahim discloses a method by where "[t]he program state and data would be transmitted . . . as a URL hidden in a hyperlink associated with the Step and Continue push-buttons." Ex. 1004 at 258; Ex. 1002 at ¶ 249.

2. '601 Patent Disclosure and Claims

The '601 Patent discloses passing state information in an HTTP communication session by a CGI script embedding state information as arguments in all URLs in a web page being generated. Ex. 1001 at Abst; Ex. 1002 at ¶¶ 30-31.

The '601 Patent includes 66 claims, six of which are independent. Independent claims 1, 14, 27, and 40 recite substantially the same elements in different formats—method steps (Claim 1), "programmable storage device" performing steps (Claim 14), "logic for" elements (Claim 27), "means for" elements (Claim 40). Similarly, Independent Claims 51 and 60 recite substantially similar elements in different formats—method steps (Claim 51), and "programmable storage device" performing steps (Claim 60).

The dependent claims depending from Independent Claim 1 substantially repeat for each independent claim; e.g., Claims 2, 15, 28, 41, 52, and 61 each substantially recite "wherein said embedding is performed by a server and said step

of communicating is in response to said embedding step."

3. Relevant Prosecution History

The '601 Patent filed on June 7, 1996. In two office action responses, Applicant argued that the invention enabled preservation of state information across multiple servers and distinguished the claims over prior art directed to maintaining state in hidden Form fields (U.S. Patent No. 5,710,918 to Lagarde) and maintaining state using cookies (U.S. Patent No. 5,774670 to Montulli). Applicant argued against a 35 U.S.C. § 112, first paragraph, rejection for non-enablement by noting that the invention was directed to and could be applied to browsing the WWW which consists of "multiple servers networked together" and HTML documents with links to documents on both local and remote servers. Ex. 1011 at p. 12. Applicant distinguished the Form disclosure by arguing that because not all web pages include Forms, this method could not preserve and embed state information during the entire life of a client-server conversation. Ex. 1011 at p. 17-18; Ex. 1002 at ¶ 42. The applicant distinguished the cookie disclosure by arguing that with the use of cookies, which are stored on the client-side and are associated with specific web addresses, there was no correlation with a specific "conversation" or "embedding the state information in all identified

continuations." Ex. 1013 at 2-4; Ex. 1002 at ¶ 44.

B. **Claim Construction (37 C.F.R. § 42.104(b)(3))**

In an *Inter Partes* Review, a claim in an unexpired patent is given the "broadest reasonable construction in light of the specification of the patent in which it appears" ("BRI"). 37 C.F.R. § 42.100(b). The following summarizes how certain claim phrases of the '601 Patent should be construed under this standard.

1. "continuation"

The BRI of "continuation" is "a new request which a client may send to a server, such as a hyperlink," which is an explicit definition in the specification. Ex. 1001 at 2:48-56; Ex. 1002 at ¶ 48-51. In the Litigation, IBM has contended that continuations are further limited to those "logically related to the original request." Ex. 1015. This is not mandated by the specification, which notes only that "useful continuations are generally logically related to the original request." Ex. 1001 at 2:55-56. A POSA would understand that this passage indicates the further limitation in IBM's construction is exemplary and not a requirement of the BRI. Ex. 1002 at ¶ 50.

2. "services"

The BRI of "services" is "programs performed by a server and invoked by a client that include as output one or more continuations." Ex. 1001 at 1:48-56; Ex. 1002 at ¶¶ 52-53.

3. "state information"

The BRI of "state information" is "information about an ongoing interaction between a client and a server during a conversation." Ex. 1002 at ¶ 54-57. The '601 Patent provides two examples of state information; first, a user ID and transaction number and, second, abstract "state variables x=32 and y=45." Ex. 1001 at 7:42-45, 15:7-12. The Perrochon article cited on the face of the '601 Patent provides an explicit definition: "Information that a server maintains about the status of ongoing interactions with clients is called state information." Ex. 1016 at p. 3. The BRI is consistent with the specification examples and this definition.

4. "conversation"

The BRI of "conversation" is "a sequence of communications between the client and server in which the server responds to each request with a set of continuations and the client always picks the next request from the set of continuations and which ends when the client does not pick a request from the set of continuations," as defined in the patent. Ex. 1001 at 7:3-9; Ex. 1002 at ¶¶ 58-60.

5. "dynamically downloading computer program code to the client to perform said step of embedding"

Independent Claims 1 and 14 recite "communicating the output" that is "in response to said step of embedding," meaning the "communicating" step must occur *after* the "embedding" step. Ex. 1002 at ¶ 61. Likewise, Independent Claims 51 and 60 recite "communicating a response including the continuations and embedded state information" generated in the embedding step, again meaning the "embedding" step has already occurred. Ex. 1002 at ¶ 62. Dependent Claims 4, 17, 54, and 63 recite "dynamically downloading . . . to perform said step of embedding ***which is responsive to said step of communicating***," meaning "said step of communicating" must occur *before* "said step of embedding," contradicting the independent claim. Ex. 1002 at ¶ 63-64. To the extent a BRI exists, any such BRI would require, for these dependent claims, ignoring the requirement of the independent claims that the embedding occurs before communicating. *Id*. at ¶ 65.

6. Terms subject to Section 112, 6

Claims 40-49 recite a number of "means for" elements, while Claims 27-38 recite parallel "logic for" elements. None of the introductory terms (e.g., "state detection") connote structure to one of ordinary skill and, thus, each of these terms is subject to Section 112, 6. Ex. 1002 at ¶¶ 66-93. Further, in most cases the

corresponding structure consists of a general purpose computer, "server 410'," performing functions (*Id.*) and, thus, the structure must also include the disclosed algorithm steps. *See Med. Instr. and Diagnosis v. Elekta*, 344 F.3d 1205, 1217-18 (Fed. Cir. 2003). In several cases, the specification does not disclose any "algorithm" beyond simply restating the function. Petitioner cites this disclosure but does not concede that it is sufficient to meet Sections 112, 6 or 112, 2.

(a) **"state detection [logic] / [means] for…"**

Function: detecting when the request for service requires preservation of the state information. Ex. 1002 at ¶ 67 Structure: Server 410' performing the algorithm set in the box labeled 510 in FIG. 4 and discussion at 11:55-58. *Id*. at ¶ 68.

(b) **"search [logic] / [means] for…"**

Function: identifying all continuations in an output from said service, in response to said step of detecting. Ex. 1002 at ¶ 71. Structure: Server 410' performing either of (a) the algorithm depicted at step 520 of FIG. 4 and discussion at 11:61-64, or (b) steps 811 and 812 of FIG. 8 and discussion at 14:29-63. *Id.* at ¶ 72.

(c) **"converter [logic] / [means] for…"**

Function: recursively embedding the state information in all identified continuations. Ex. 1002 at ¶ 75. Structure: Server 410' performing either of (a) the

algorithm depicted at step 520 of FIG. 4 and discussion at 11:61-64, or (b) steps 811 and 812 of FIG. 8 and discussion at 14:29-63. *Id.* at ¶ 76.

(d) **"communication [logic] / [means] for…"**

Function: communicating the output to the client. Ex. 1002 at ¶ 79. Structure: Server 410' performing the algorithm discussed at 12:45-48. *Id.* at ¶ 80.

(e) **"means for modifying…" [Claim 48]**

Function: modifying an identified continuation which is a request for an HTML file to invoke a CGI converter program with the identified continuation and the state information passed as arguments. Ex. 1002 at ¶ 83. Structure: Server 410' performing step 811 of FIG. 8 and discussion at 14:29-63. *Id.* at ¶ 84.

(f) **"means for modifying…" [Claim 49]**

Function: modifying an identified continuation which is an invocation to a CGI program with the identified continuation and the state information passed as arguments. Ex. 1002 at ¶ 87. Structure: Server 410' performing step 812 of FIG. 8 and discussion at 14:29-63. *Id.* at 88.

C. **Level of Ordinary Skill in the Art**

A person of ordinary skill in the relevant art ("POSA") at the time the '601 Patent was filed (June 7, 1996) would have had at least a bachelor of science degree in electrical engineering or computer science, or a master's degree in

information science. The POSA would also have two to three years of engineering experience related to distributed systems or network applications. Ex. 1002 at ¶ 17.

D. **The Challenged Claims Are Invalid Under §§ 102(a) or 103(a)**

Pursuant to 37 C.F.R. § 42.104(b)(4)-(5), the following analysis and evidence demonstrates where each element of the Challenged Claims is found in the prior art for each of the grounds listed above.

1. Ground 1 – Claims 1, 2, 6-8, 11, 12, 14, 15, 20, 21, 25, 27, 28, 32, 33, 34, 37, 38, 40, 41, 45, 47-49, 51, 52, 56, 57, 60, 61, 65, and 66 are rendered obvious by Graber

(a) **Overview of Graber**

Like the ’601 Patent, which encompasses preserving state in communications with “one or more servers,” and which uses the WWW as an example (Ex. 1001, at 9:43-52), Graber is directed to preserving state information during a user web session across multiple servers, particularly the navigational history of a user on the WWW, and including the identity of referring websites that directed a user to a later site. Ex. 1003 at Abstract. Graber frames the problem as follows: “when a user navigates through various sites during a session on the Web, the navigational history reflecting the past locations traversed is typically lost, meaning it is difficult for the user or for any service monitoring the user to know

the identity of any previous site traversed by the user during a session." Ex. 1003 at 1:45-54. A particular object of the invention claimed by Graber is capturing and tracking a co-marketing website that directed a new subscriber to an on-line service, particularly without requiring "participation or intervention from the new subscriber." Ex. 1003 at 1:41-44. Graber accomplishes its preservation of state across different servers by using the same technique disclosed in the '601 Patent, namely, it embeds state information (in Graber's case, a code representing the identity of the directing website) into the URLs visited by a user as he traverses the WWW. Ex. 1003 2:65-3:5. This is done by including a string in the URL which calls a CGI program. The CGI program takes as arguments, an identifier representing the web page to which the user has been directed and a code associated with the previous referring website. Ex. 1003 at 11:54-66.

For example, in the URL "WWW.OLS.COM\page_1ink.cgi?index@CM1," the first portion (i.e., WWW.OLS.COM) identifies the web site to which the user is being directed. The remaining portion (i.e., page_link.cgi?index@CM1) represents a call to the page_link.cgi program with two arguments, namely, a destination page identifier (i.e., index) representing the particular page at the site to which the user

has been directed, and a code (i.e., CM1) representing the identity of the web site that directed the user to the site. Ex. 1003 at 11:66-12:16.

The specification and the accompanying figure explain the operation of the page_link.cgi program. In particular, when the web server receives the URL which includes a string containing a call to the page_link.cgi program, the program first extracts the destination page identifier (in our example "index") and a code or identifier for the referring site (in our case "CM1"). *Id*. at 12:22-25. It next retrieves the page represented by the page identifier and selects and tests each hyperlink on the page in order to determine whether the URL includes further calls to page_link.cgi. *Id*. at 12:26-34. For all links on a page that includes a call to page_link.cgi, the program appends the code for the referring page to the end of the URL. The destination page, which now includes URLs having the appended code, is then passed back to the user. *Id*. at 12: 34-56. When the user desires to move off the destination page by selecting another of the URL page links which contains a call to page_link.cgi, the process is repeated. *Id*. at 12:56-63.

Accordingly, the identity of an original website from which the user navigated is preserved throughout a web session as the user continues to select links that contain a call to the page_link.cgi function. Notably, where the user

selects a page link with a call to page_link.cgi which links to an external URL, the page_link.cgi program (in addition to embedding the code for an original referring website) also appends a code representing the current website. In that way, several codes representing websites traversed during a user session may be stored in the URL. FIG. 6 illustrates the method:

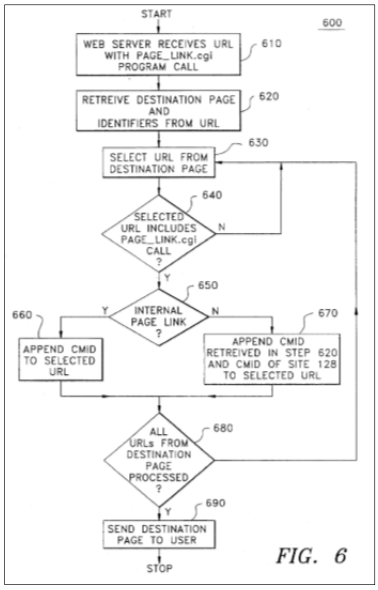

FIG. 6

The specification describes the preferred embodiment of the above algorithm being the case where each page link URL on a web page will include a call to the page_link.cgi program, where the URLs on a web page direct the user to another web page at the same site or to any page that can recognize the identifiers for originating sites. *See Id.* at 13:46-53. This, explains the specification, insures that information about the original site from which the user was directed is available to

future websites visited by the user. *Id*. at 13:58-64.

(b) **Independent method claims**

(1) ***Claim 1***

1.1. "A computerized method for preserving state information in a conversation between a client adapted to request services from one or more servers which are networked via a stateless protocol to the client."

To the extent the preamble is limiting, Graber teaches this limitation because Graber discloses a **method operated by a computer** web server of **preserving** navigation history of a client by appending the code for referring pages to the end of URLs requested by the user when generating a new HTML page in response to a user **request** to one or more servers. Ex. 1003 at 11:57 (URL . . . includes a string which functions to call a special page_link.cgi program which runs on web server 142."); 12:43-53 ("[F]or each internal URL in the destination page which includes a string for calling the page_link.cgi program, the page_link.cgi program appends the UNIX symbolic link/CMID code (i.e. CM1) originally passed as an argument to the program end of the URL. In addition, in step 670, for each external URL in the destination page which includes the 'page_link.cgi' string, the page_link.cgi program appends the UNIX symbolic link/CMID originally passed to the program followed by a UNIX symbolic link/CMID representing OLS site 128 (e.g., IOLS) to the end of the URL."); Ex. 1002 at ¶ 97.

These HTML pages constitute data delivered by the server and invoked by a client that include as output one or more continuations (in the form of hyperlinks). Ex. 1003, Claim 1 ("wherein said first WWW web site has a first plurality of web pages"); Ex. 1003 at 12:38-32 ("Each page at web site 128 is represented by a file which includes one or more fields containing further URLs representing links to other pages at web site 128 (internal page links) or to pages at web sites other than site 128 (external page links)."); Ex. 1002 at ¶ 98. The web sites are therefore "**services**" under the BRI. Ex. 1002 at ¶ 98.

The user's navigation history information is carried forward **throughout the user session** described by Graber, which is a user's navigation "through various sites during a session on the WWW." Ex. 1003 at 1:45-48; Ex. 1002 at ¶ 99. Such a user session is a sequence of communications between the client and servers in which a server responds to each request with a set of continuations where the client may select the next request from the set of continuations. Ex. 1003 at 12:28-32 ("Each page at web site 128 is represented by a file which includes one or more fields containing further URLs representing links to other pages at web site 128 (internal page links) or to pages at web sites other than site 128 (external page links)."); 5:28-33 ("In the preferred embodiment of the present invention, a page at

site 122a includes an advertisement (not shown) for OLS 140. In addition, the advertisement at co-marketer site 122a is preferably such that a user of user station 102a may choose to connect to OLS site 128 simply by 'clicking' on the advertisement at WWW site 122a."); Ex. 1002 at ¶ 99. This is precisely the type of navigation through numerous local and remote servers that the applicant argued his invention enabled. Ex. 1011 at p. 14 ("continuations include links to "other servers"). It is therefore a "**conversation**" under the BRI. Ex. 1002 at ¶ 99.

The navigational history information about the user traversing the WWW is information about the ongoing interaction between a client and a server during a conversation; and it is, therefore, "**state information**" under the BRI. Ex. 1002, at ¶ 100. To the extent IBM argues that navigation history is not state information, a POSA would find it obvious to use the same technique with other types of information that would constitute state information. Such information may include the user identification number that Graber teaches will be assigned to each user, Ex. 1003 at 2:43 ("The enrollment means further includes means for storing . . . the unique identification number of a user . . . ."), given Graber's stated goal of delivering on-line services which are accessed by user subscribers. Ex. 1003 at 1:16-20 (giving "on-line information retrieval services, on-line travel reservation

services, or on-line stock trading services" as examples); Ex. 1002 at ¶ 100. A POSA would understand that such information could be useful to the method and apparatus taught by Graber and thus would be motivated to modify or add to the type of information passed in the URL to further Graber's goals. A user identification number, for instance, would be useful for purposes of tracking specific users, preventing repeated subscriptions, or granting access to protected resources. *Id*. The '601 Patent explicitly gives a user identification number as an example of state information. Ex. 1001 at 7:42-45, 15:7-12. Graber discloses communicating on the WWW using HTTP, which represents a collection of servers **networked via a stateless protocol**. *See e.g.*, Ex. [Graber] at Table II ("specifying URL as http://www.ols.com/cm1/subdir1/subdir2/subdir3/page.htm.).

1.2: "said services including one or more of data and programs which the client may request, wherein the conversation is a sequence of communications between the client and one or more servers for said services wherein each response from the server includes one or more continuations which enable another request for said services and wherein the client must invoke one of the continuations to continue the conversation"

Graber teaches **services including one or more data and programs which the client may request** because Graber describes that each response from the server includes a request for a webpage with one or more hyperlinks. Ex. 1003 at 12:38-32 ("Each page at web site 128 is represented by a file which includes one or

more fields containing further URLs representing links to other pages at web site 128 (internal page links) or to pages at web sites other than site 128 (external page links).”). As noted above, these web pages constitute data delivered by servers and invoked by a client that include as output one or more continuations (in the form of hyperlinks). Ex. 1003, Claim 1 (“wherein said first WWW web site has a first plurality of web pages”). The web pages are therefore “services” under the BRI.

Graber further teaches **wherein the client must invoke one of the continuations to continue the conversation** because Graber discloses a navigation history created by a user “clicking” on, for instance, an advertisement at a WWW site. *Id*. at 5:23-33.

1.3: “the client initiating the conversation with the server using the stateless protocol”

Graber teaches this limitation because Graber discloses a user initiating a session (i.e. a conversation) so as to “navigate[] through various sites during a session on the WWW.” Ex. 1003 at 1:45-48. To the extent there is no explicit disclosure of the user initiating the session, a POSA would understand that this step is inherent because a session defines a user’s navigation path and it is the user who directs such navigation, including its initiation. Ex. 1002 at ¶ 106 To the extent this claim limitation is not disclosed, it would been obvious to a POSA to

have the client initiate the browsing session because one of either the client or server must begin the conversation between the participants. *Id*. Graber discloses communicating using HTTP, which is a stateless protocol. Ex. 1003 at 13:1-45.

1.4: "detecting when the request for a service requires preservation of the state information"

Graber teaches this limitation because Graber describes delivering to a web server a URL which includes a string containing a call to "page_link.cgi." The server's receipt of a URL including a call to this CGI program indicates that the request for service will require preservation of the state information and the invoked program call indeed preserves and passes forward state information (in the form of a code associated with an originating website) to future websites for the duration of a user session. *Id*. at Fig. 6; 12:19-23 ("In step 610, when web server 142 receives a URL which includes a string containing a call to the page_link.cgi program, the page_link.cgi program is invoked on the web server 142"); 13:53-64 ("By inserting the page_link.cgi program call and [a referring site code] into each page link that points to a further web page at site 128, . . . the system permits the user to carry [the code] to further web sites.").

1.5: "performing said service and identifying all continuations in an output from said service, in response to said step of detecting"

Graber teaches the **performing said service and identifying all continuations in an output from said service** limitation because Graber discloses "retriev[ing] [a] destination page." *See Id*. at Fig. 6, step 620; 12:59-61. This is the web page to which the user has been directed by the previous website and reflects performance of a service under the BRI. The web page contains one or more hyperlinks representing links to other pages on the WWW. *Id*. at 12:28-32. Graber further identifies all continuations, i.e. hyperlinks, in the output, because it must select and test each of the links. *Id*. at 12:31-34 ("Each URL in the destination page is then selected and tested (in steps 630 and 640) in order to determine whether the URL includes a string for calling the page_link.cgi program.").

Further, Graber discloses the performing and identifying step **in response to said step of detecting**. Specifically, Graber takes the identifying step after and only where there is a call to the page_link.cgi program, the presence of which allows the server to detect when the request for a service requires preservation of the state information. *Id*. at 12:19-34 ("In step 610, when web server 142 receives a URL which includes a string containing a call to the page_link.cgi program, the page_link.cgi program is invoked on the web server 142. * * * Each URL in the destination page is then selected and tested (in steps 630 and 640) in order to

determine whether the URL includes a string for calling the page_link.cgi program.").

1.6: "recursively embedding the state information in all identified continuations"

Graber teaches this limitation because Graber discloses embedding a code referencing a referring website, i.e. state information, in all of the hyperlinks on a website where the links point to a further page at the site or any other website which can recognize the referring site code that has been embedded into the URL by iterating through all page links. *Id*. at FIG. 6; 12:43-53 ("for each internal URL in the destination page which includes a string for calling the page_link.cgi program, the page_link.cgi program appends the [a code for the referring website] originally passed as an argument to the program to the end of the URL. In addition, in step 670 for each external URL in the destination page which includes the "page_link.cgi" string, the page_link.cgi program appends the [a code for the referring website] originally passed to the program followed by a [code] representing [the current site] to the end of the URL."); 13:45-55 ("In the embodiment shown in FIG. 6 and described above, each page link URL on a web page at site 128 will preferably include a call to the page_link.cgi program if the page link points to either (i) a further web page at site 128, or (ii) a further web site

which is adapted to recognize UNIX symbolic links that have been inserted into a URL by a previous web site during a user session."); Ex. 1002 at ¶ 112. Accordingly, Graber discloses making state information part of all the continuations identified in the output by the server. *Id*.

1.7: "communicating the output to the client, in response to said step of embedding, wherein the state information is preserved and provided to all services for the duration of the conversation"

Graber teaches **communicating the output to the client in response to said step of embedding** because Graber describes passing the destination page back to the user which includes URLs having the appended codes for the referring website. The communicating step is **in response to said step of embedding** because the step is taken after and only as a result of the execution of the page_link.cgi program. *Id*. at 12:54-56 ("The destination page, which includes URLs having the appended codes described above is then passed back to the user in step 690."); Fig. 6; Ex. 1002, at ¶ 114.

Graber further teaches **state information is preserved and provided to all services for the duration of the conversation** because Graber explains that as a result of the page_link.cgi program call and its function of embedding a code for a referring site into a URL, the user's navigation history will be provided as part of

the URL for all future websites traversed for the duration of the sequence of communications between the client and service. *Id*. at 13:53-64; Ex. 1002 at ¶ 115.

(2) ***Claim 51***

51:1: "A computerized method for preserving state information in a conversation via a stateless protocol between a client adapted to request services from one or more servers, the method comprising the steps of:

To the extent the preamble is limiting, Graber discloses this element for at least the reasons of element. 1.1. Ex. 1002 at ¶ 117

51.2 receiving a service request including state information, via the stateless protocol;

Graber discloses this element for at least the reasons of element 1.3. Ex. 1002 at ¶ 118

51.3 identifying all continuations in an output from said service and recursively embedding the state information in all identified continuations, in response to said request; and

Graber discloses this element for at least the reasons of elements 1.5 and 1.6. Ex. 1002 at ¶ 119.

51.4 communicating a response including the continuations and embedded state information, wherein the continuations enable another service request and one of the continuations must be invoked to continue the conversation.

Graber discloses this element for at least the reasons of element 1.7. Ex. 1002 at ¶ 120

(c) **Independent "program storage device" claims**

(1) ***Claim 14***

14.1: "A program storage device readable by a computer, tangibly embodying a program of instructions executable by the computer to provide a method for preserving state information in a conversation between a client adapted to request services from one or more servers which are networked via a stateless protocol to the client,

To the extent the preamble is limiting, Graber discloses and renders obvious this claim element for at least the reasons of element 1.1. Further, Graber teaches a program storage device, in the form of the memory of a web server 142 readable by a computer embodying a special page_link.cgi program, which are instructions executable for performing the steps set forth in claim elements 1.1-1.7. Specifically, Graber discloses that this functionality is performed by a user station 102a and web server 142. *See* Ex. 1003 at 5:14-18; 11:54-58; Ex. 1002 at 121.

14.2 "said services including one or more of data and programs which the client may request, wherein the conversation is a sequence of communications between the client and one or more servers for said services wherein each response from the server includes one or more continuations which enable another request for said services and wherein the client must invoke one of the continuations to continue the conversation, the method comprising the steps of:"

14.3: "the client initiating the conversation with the server using the stateless protocol"

14.4 detecting when the request for a service requires preservation of the state information;

14.5 performing said service and identifying all continuations in an output from said service, in response to said step of detecting;

14.6 recursively embedding the state information in all identified continuation; and

14.7 communicating the output to the client, in response to said step of embedding; wherein the state information is preserved and provided to all services for the duration of the conversation.

Graber teaches and renders obvious elements 14.2-14.7 for at least the reasons discussed above for claim elements 1.2-1.7, respectively, and a POSA would understand that software running on the memory of a web server as discussed with regard to element 14.1 would perform those elements for at least the reasons of 1.2-17, respectively. Ex. 1002 at ¶¶ 122-127.

(2) ***Claim 60***

60.1: "A program storage device readable by a computer, tangibly embodying a program of instructions executable by the computer to provide a method for preserving state information in a conversation via a stateless protocol between a client adapted to request services from one or more servers, the method comprising the steps of:

To the extent the preamble is limiting, Graber discloses and renders obvious this element for at least the reasons of element 14.1. Ex. 1002 at ¶ 128.

60.2 receiving a service request including state information, via the stateless protocol;

Graber discloses and renders obvious this element for at least the reasons of element 14.3. Ex. 1002 at ¶ 129.

60.3 identifying all continuations in an output from said service and recursively embedding the state information in all identified continuations, in response to said request;

Graber discloses and renders obvious this element for at least the reasons of element 14.5 and 14.6. Ex. 1002 at ¶ 130.

60.4 and communicating a response including the continuations and embedded state information, wherein the continuations enable another service request and one of the continuations must be invoked to continue the conversation.

Graber discloses and renders obvious this claim element for at least the reasons of element 14.7. Ex. 1002 at ¶ 131.

(d) **Independent claims governed by 35 U.S.C. § 112, 6**

(1) ***Claim 27***

(a) ***27.1: "A computer system for preserving state information in a conversation between a client adapted to request services from one or more servers which are networked via a stateless protocol to the client"***

To the extent this preamble is limiting, Graber discloses and renders obvious this element at least for at least the reasons of element 1.1. Further, Graber discloses that the functionality discussed with regard to Claim 1 is performed by a server and client, which are collectively the claimed "computer system." *See* Ex. 1003 at 4:17-21; Fig. 6; Ex. 1002 at ¶ 132.

(b) ***27.2: "said services including one or more of data and programs which the client may request, wherein the conversation is a sequence of communications between the client and one or more servers for said services, wherein each***

response from the server includes one or more continuations which enable another request for said services and wherein the client must invoke one of the continuations to continue the conversation, the system comprising"

Graber discloses and renders obvious this element for at least the reasons of element 1.2. Ex. 1002 at ¶ 133.

(c) ***27.3: "the client being adapted for initiating a conversation with the server using the stateless protocol"***

Graber discloses and renders obvious this element for at least the reasons of element 1.3. Ex. 1002 at ¶ 134.

(d) ***27.4: "state detection logic for detecting when the request for a service requires preservation of the state information"***

Graber discloses and renders obvious this element for at least the reasons of element 1.4. The functionality of 1.4 meets this function. The structure is web server 142 performing the algorithm of element 1.4 and particularly disclosed at Fig. 6, box 640. Ex. 1002 at ¶ 135.

(e) ***27.5: "search logic for identifying all continuations in an output from said service, in response to said step of detecting"***

Graber discloses and renders obvious this element for at least the reasons of element 1.5. The functionality of 1.5 meets this function. The structure is web server 142 performing the algorithm of 1.5 and particularly disclosed at Fig. 6, in boxes 630 and 680. Ex. 1002 at ¶ 136.

(f) ***27.6: "converter logic for recursively embedding the state information in all identified continuations"***

Graber discloses and renders obvious this element for at least the reasons of element 1.6. The functionality of element 1.6 meets the function, and the structure is web server 142 performing the algorithm of claim element 1.6 and particularly disclosed at Fig. 6, box 660 and 670. Ex. 1002 at ¶ 137.

(g) ***27.7 "communication logic for communicating the output to the client; wherein the state information is preserved and provided to all services for the duration of the conversation"***

Graber teaches claim element 27.7 for at least the reasons discussed above for claim element 1.7. The functionality of element 1.7 meets this function. The structure is web server 142 performing the algorithm of element 1.7 and particularly disclosed at Fig. 6 at box 690. Ex. 1002 at ¶ 138.

(2) ***Claim 40***

(a) ***40.1: "A computer system for preserving state information in a conversation between a client adapted to request services from one or more servers which are networked via a stateless protocol to the client"***

(b) ***40.2: "said services including one or more of data and programs which the client may request, wherein the conversation is a sequence of communications between the client and one or more servers for said services wherein each response from the server includes one or more continuations which enable another request for said services and wherein the client must invoke one of the continuations to continue the conversation, the system comprising"***

(c) ***40.3: "the client being adapted for initiating the conversation with the***

server using the stateless protocol"

(d) ***40.4 state detection means for detecting when the request for a service requires preservation of the state information"***

(e) ***40.5 search means for identifying all continuations in an output from said service, in response to said step of detecting"***

(f) ***40.6 converter means for recursively embedding the state information in all identified continuations"***

(g) ***40.7 communication means for communicating the output to the client; wherein the state information is preserved and provided to all services for the duration of the conversation"***

Graber teaches elements 40.1-40.7 for at least the reasons discussed above for claim elements 27.1-27.7. Ex. 1002 at ¶¶ 139-145.

(e) **Dependent Claims**

(1) ***Claims 2, 15, 28, 41, 52, 61***

Claim 2, 15, 28, 41, 52, and 61 depends from Claims 1, 14, 27, 40, 51, and 60 respectively, and as shown above, Graber discloses and renders obvious every element of Claims 1, 14, 27, 40, 51, and 60. Ex. 1002 at ¶¶ 146-161.

Further, each recite substantially the same language:

(a) ***wherein, the step of embedding is performed by the server and said step of communicating is in response to said step of embedding."***

Graber teaches the step of **embedding is performed by the server** because Graber discloses that the page_link.cgi program—which as shown above is

responsible for the embedding step—is invoked on the web server. Ex. 1003 at 12:19-22 ("In step 610, when web server 142 receives a URL which includes a string containing a call to the page_link.cgi program, the page_link.cgi program is invoked on the web server 142.").

Graber further teaches said step of **communicating is in response to said step of embedding** because Graber describes a communicating step occurring in response to and following the step of embedding codes for referring websites in the URLs. Namely, the resulting web page is passed back to the user after the page_link.cgi program has been invoked and has embedded codes for a referring website to the end of the URLs on the resulting web page. *Id*. at 12:54–56 ("The destination page, which includes URLs having the appended codes described above is then passed back to the user in step 690.").

Further, with regard to Claims 15 and 61, Graber discloses "program storage device readable by a computer tangibly embodying a program of instructions executable by the computer" for performing the functionality discussed above, namely code running on web server 142 as in Claim 14. Ex. 1002 at ¶¶ 152, 161. And, with regard to Claims 28 and 41, the functionality discussed above meets the

function, and the structure is web server 142 performing the algorithm discussed above and particularly disclosed in Fig. 6 at box 690. *Id.* at ¶¶ 154, 156.

(2) ***Claims 6, 19, 56, 65***

Claims 6, 19, 56, and 65 depend from Claims 1, 14, 51, and 60 respectively, and as shown above, Graber discloses and renders obvious every element of Claims 1, 14, 51, and 60. Ex. 1002 at ¶¶ 162-188.

Further, each of Claims 6, 19, 56, 65 recite substantially the same language:

(a) ***"the client selecting a second continuation from said all identified continuations with embedded state information"***

Graber teaches this limitation because after a client has navigated by clicking a first URL page link with a call to the page_link.cgi program, the resulting page passed back to the user includes URLs having the appended or embedded codes for the referring web site. Ex. 1003 at 12:54-58 ("The destination page, which includes URLs having the appended codes described above is then passed back to the user in step 690."). As discussed above with respect to element 1.1, these codes are state information under the BRI or it would have been obvious to use the same technique with other types of information that would constitute state information. Ex. 1002 at ¶ 100. Graber discloses further calls to the page_link.cgi program in the destination page returned to the user. Where a user

selects a second URL page link with another call to the page_link.cgi program, the link contains the embedded state information. Ex. 1003 at 12:54-64 ("The destination page, which includes URLs having the appended codes described above is then passed back to the user in step 690. Thereafter, when the user desires to move off of the destination page (passed to the user in step 690), the user will select one of the URL page links on the user's page as a new destination page. If the URL corresponding to this new destination page contains a call to the page_link.cgi program described above, the process described above is repeated from step 610 using the URL of the new destination."). Ex. 1002 ¶¶ 163.

Further, with regard to Claims 19, Graber includes a "program storage device readable by a computer tangibly embodying a program of instructions executable by the computer" for performing the functionality of this claim element, namely code running on a user station 102a as shown with regard to Claim 14. *Id.* at ¶ 168.

With regard to Claim 65, which recites: "receiving a second request associated with a second continuation . . .," Graber discloses stored code for performing the functionality of this claim element, namely code running on a web server 142 as shown with regard to Claim 14. *Id*. at ¶ 186.

(b) ***"restoring the state information from said second continuation and invoking an associated second service with restored state"***

Graber teaches this limitation because where a user selects a second URL page link with another call to the page_link.cgi program, the link contains the embedded state information and the process of extracting and embedding the code for the originating site to the destination page is repeated. *Id*. at 12:54-64 ("The destination page, which includes URLs having the appended codes described above is then passed back to the user in step 690. Thereafter, when the user desires to move off of the destination page (passed to the user in step 690), the user will select one of the URL page links on the user's page as a new destination page. If the URL corresponding to this new destination page contains a call to the page_link.cgi program described above, the process described above is repeated from step 610 using the URL of the new destination."); Ex. 1002 at ¶ 164.

Further, with regard to Claims 19 and 65, Graber includes a "program storage device readable by a computer tangibly embodying a program of instructions executable by the computer" for performing the functionality of this claim element, namely code running on a web server 142 as shown with regard to Claim 14. *Id.* at ¶¶ 169, 187.

(c) ***"recursively identifying and embedding the state information in all continuations associated with an output from said second service"***

Graber teaches this limitation because Graber discloses identifying state information using the page_link.cgi program which extracts a code representing the identity of a referring website. Ex. 1003 at 12:23-26 ("Next, in step 620, the page_link.cgi program extracts the destination page identifier (e.g., index) and UNIX symbolic link/CMID code (e.g.. CM1) that were contained as arguments in the page_link.cgi program call."). Ex. 1002 at ¶ 165. Graber describes a preferred embodiment where the code for the referring site, which is state information under the BRI (or as discussed above with respect to element 1.1, it would have been obvious to use the same technique with other types of information that would constitute state information, Ex. 1002 at ¶ 100) is embedded in all the page link URLs on the web page site (the output from the second service) where the page links point to either (i) a further web page at the same site or (ii) a further web site which can recognize the code for a referring website. Ex. 1003 at 13:45-53 ("In the embodiment shown in FIG. 6 and described above, each page link URL on a web page at site 128 will preferably include a call to the page_link.cgi program if the page link points to either (i) a further web page at site 128, or (ii) a further web site which is adapted to recognize UNIX symbolic links that have been inserted into a

URL by a previous web site during a user session."); Ex. 1002 at ¶ 166. Further, with regard to Claims 19 and 65, Graber includes a "program storage device readable by a computer tangibly embodying a program of instructions executable by the computer" for performing the functionality of this claim element, namely code running on a web server 142 as shown with regard to Claim 14. *Id.* at ¶¶ 171, 189.

(3) ***Claims 32 and 45***

Claims 32 and 45 depend from Claims 27 and 40 respectively, and as shown above, Graber discloses every element of Claims 27 and 40. Further, each of Claims 32 and 45 recite substantially the same language:

(a) ***the converter [logic] / [means] being further adapted for restoring the state information from said second continuation, invoking an associated second service with restored state information, and recursively identifying and embedding the state information in all continuations associated with an output from said second service***

Graber discloses and renders obvious this claim element for at least the reasons shown with regard to elements 6.2 and 6.3. Ex. 1002 at ¶¶ 174, 175. The functionality shown in elements 6.2 and 6.3 meets the function, and the structure is the web server 142 performing the algorithm of Fig. 6 for a first and second destination page selected by the user that a contains a call to page_link.cgi. Ex. 1003:56-64 ("Thereafter, when the user desires to move off of the destination page

(passed to the user in step 690), the user will select one of the URL page links on the user's page as a new destination page. If the URL corresponding to this new destination page contains a call to the page_link.cgi program described above, the process described above is repeated from step 610 using the URL of the new destination."); Ex. 1002 at ¶¶ 174-175.

(4) ***Claims 7, 20, 33***

Claims 7, 20, and 33 depend from Claims 1, 14, and 27 respectively, and as shown above, Graber discloses and renders obvious every element of Claims 1, 14, and 27. Further, each recite substantially the same language:

(a) ***"further comprising the step of correlating the state information to a specific conversation"***

Graber teaches this limitation because Graber discloses tracking the navigation path of a user during a particular session on the WWW and correlating the identity of a referring websites (which comprises state information under the BRI or as discussed with respect to element 1.1, it would have been obvious to use the same technique with other types of information that would constitute state information, Ex. 1002 at ¶ 100) with that particular session. Ex. 1003 at 1:45-53 ("When a user navigates through various sites during a *session* on the WWW, the navigational history reflecting the *past locations traversed by the user during the*

session is typically lost as the user moves from one site to the next site. Thus, unless the user were to manually track the various sites traversed *during a world wide web session*, it would be difficult for the user, or for any service monitoring the user, to know the identity of any previous world wide web site traversed by the user during a session.") (emphasis added). A session as disclosed by Graber is "a sequence of communications between the client and server in which the server responds to each request with a set of continuations and the client always picks the next request from the set of continuations and which ends when the client does not pick a request from the set of continuations" because Graber describes a user session as a user traversing URL page links. *See* Ex. 1003 at 13:58-64 ("In addition, by inserting the UNIX symbolic link information associated with both a previous web site 122a, b, c and OLS site 140 into page links associated with different web sites (other than site 128), the system permits the user to carry UNIX symbolic link information representing previous location(s) traversed during a user session to further web sites."); Ex. 1002 at ¶ 191. This is precisely the type of navigation through numerous local and remote servers that the applicant argued the invention enabled. Ex. 1011 at p. 14 ("continuations include links to 'other servers'"); Ex. 1002 at ¶ 100.

For example, a user my select or click on an advertisement on a particular page. Ex. 1003 at 5:30-33 ("In addition, the advertisement at co-marketer site 122a is preferably such that a user of user station 102a may [sic] chose to connect to OLS site 128 simply by 'clicking' on the advertisement at WWW site 122a."). Accordingly, a "user session" as described in Graber is a "conversation" under the BRI. Ex. 1002 at ¶ 191. As discussed above, the navigation information stored in the URLs by the system of Graber is correlated to the specific ongoing user session. Ex. 1003 at 1:45-53; Ex. 1002 at ¶ 191.

Further, with regard to Claim 20, Graber discloses "program storage device readable by a computer tangibly embodying a program of instructions executable by the computer" for performing the functionality discussed above, namely code running on web server 142 as in Claim 14. *Id.* at ¶ 194. And, with regard to Claim 33, the functionality discussed above meets the function, and the structure is web server 142 performing the algorithm in Fig. 6. *Id.* at ¶ 197.

(5) ***Claims 8, 21, 34, 47, 57, and 66***

Claims 8, 21, 34, 47, 57, and 66 depend from Claims 1, 14, 27, 40, 51, and 60 respectively, and as shown above, Graber discloses and renders obvious every

element of Claims 1, 14, 27, 40, 51, and 60. Ex. 1002 at ¶ 198-221. Further, each recite substantially the same language:

(a) ***"wherein the client and the server are networked via the World Wide Web"***

Graber teaches this limitation because Graber is directed toward tracking the navigation path of a user from a first site on the World Wide Web to a second site. Ex. 1003 at Abstract ("A method and apparatus for tracking the navigation path of a user that has been directed to a second site on the World Wide Web (WWW) from a first site on the WWW."); Ex. 1002 at ¶199.

(b) ***"the stateless protocol is hypertext transfer protocol"***

Graber discloses the stateless protocol is hypertext transfer protocol. Ex. 1003, at 13:1-47, Table II ("specifying URL as http://www.ols.com/cm1/subdir1/subdir2/subdir3/page.htm.). Ex. 1002 at ¶ 200.

(c) ***"The continuations are hyperlinks to one of hypertext markup language files and common gateway interface programs."***

Graber teaches this limitation because Graber discloses website pages on the WWW with URLs representing links to other pages (which are hypertext markup language files) at the website and URLs with calls to CGI programs, including one denoted by "page_link.cgi." Ex. 1003 at 12:27-34; Ex. 1002 at ¶ 201.

(6) ***Claims 11, 24, 37***

Claims 11, 24, and 37 depends from claims 8, 21, and 47 respectively, and as shown above, Graber discloses and renders obvious every element of Claims 8, 21, and 34. Ex. 1002 at ¶ 222-230.

Further, each of claims 11, 24, and 37 recite substantially the same language:

(a) ***"wherein said step of embedding further comprises the step of: modifying an identified continuation which is a request for an HTML file to invoke a CGI converter program"***

Graber discloses selecting and testing a URL requesting an HTML file—such URL being a continuation as described in the '601 Patent. Ex. 1001 at 6:59-60 ("Recall that hypertext links are examples of 'continuations' in client-server communication.")—to determine whether the URL has a string containing a call to a specific CGI program, denoted by the name "page_link.cgi." Ex. 1003 at Fig. 6; 12:5 (disclosing URL requesting the following HTML file: "WWW.OLS.COM\page_link.cgi?index@CM1"). Where the URL has the **identified** the "page_link.cgi" string, the URL is **modified** so as to append the code representing a referring website (which is state information under the BRI or as discussed above with respect to element 1.1, it would have been obvious to use the same technique with other types of information that would constitute state information, Ex. 1002 at ¶ 100) to the end of the URL. The resulting modified

URL invokes a CGI program to convert the URL, namely the page_link.cgi converter program. Ex. 1003 at Fig. 6; 12:43-48 ("Next, in step 660, for each internal URL in the destination page which includes a string for calling the page_link.cgi program, the page_link.cgi program appends the UNIX symbolic link/CMID code (i.e. CM1) originally passed as an argument to the program to the end of the URL."); Ex. 1002 at ¶ 223.

(b) ***"with the identified continuation and the state information passed as arguments"***

The modified URL contains a call to page_link.cgi and takes two arguments: (1) a destination page identifier representing the particular page at a site to which the user has been directed, which is the identified continuation, and (2) a code representing the identity of the web site that directed the user to the site (i.e. the state information). Ex. 1003 at 12:9-16 ("The [page_link.cgi] program call includes two arguments, namely, a destination page identifier (i.e., index) representing the particular page at site 128 to which the user has been directed, and a UNIX symbolic link/CMID code (i.e., CM1) representing the identity of the web site 122a that directed the user to site 128."); Ex. 1002 at ¶ 224.

(7) ***Claim 48***

Claim 48 depends from Claim 47 and, as shown above, Graber discloses and renders obvious Claim 47. Ex. 1002 at ¶ 231. In addition, Claim 48 recites the following:

(a) ***"wherein said converter means further comprises: means for modifying an identified continuation which is a request for an HTML file to invoke a CGI converter program with the identified continuation and the state information passed as arguments"***

Graber discloses this claim element for at least the reasons of Claim 11. Ex. 1002 at ¶ 232. The functionality of Claim 11 meets the function, and the structure is web server 142 performing the algorithm disclosed at Fig. 6 at boxes 630, 660, and 670. Ex. 1002 at ¶ 232.

(8) ***Claims 12, 25, 38***

Claims 12, 25, and 38 depends from Claims 8, 21, and 34, respectively, and as shown above, Graber discloses and renders obvious every element of Claims 8, 21, and 34. Ex. 1002 at ¶¶ 234-245. Further, each of the Claims 12, 25, and 38 recite substantially the same language:

(a) ***wherein said step of embedding further comprises the step of: modifying an identified continuation which is an invocation to a CGI program,***

Graber teaches this limitation because Graber discloses selecting and testing a URL invoking a CGI program (such URL being a continuation as described in

the '601 Patent, Ex. 1001 at 6:59-60 "Recall that hypertext links are examples of 'continuations' in client-server communication.") to determine whether the URL has a call to a specific CGI program, denoted by the name "page_link.cgi." Ex. 1003 at 12:5 (disclosing testing of URL "WWW. OLS.COM\page_link.cgi?index@CM1," which includes invocation of CGI program). Where the URL is **identified** as having the "page_link.cgi" string, it is **modified** so as to append the code representing a referring website (which is state information under the BRI or as discussed with respect to element 1.1, it would have been obvious to use the same technique with other types of information that would constitute state information, Ex. 1002 at ¶ 100) to the end of the URL. Ex. 1003 at 12:43-48 ("Next, in step 660, for each internal URL in the destination page which includes a string for calling the page_link.cgi program, the page_link.cgi program appends the UNIX symbolic link/CMID code (i.e. CM1) originally passed as an argument to the program to the end of the URL."); Ex. 1002 at ¶ 235.

(b) ***"with the identified continuation and the state information passed as arguments"***

The modified URL contains a call to page_link.cgi and that CGI program takes two arguments: (1) a destination page identifier representing the particular page at a site to which the user has been directed, which is the identified

continuation (Ex. 1001 at 6:59-60 "Recall that hypertext links are examples of 'continuations' in client-server communication."), and (2) a code representing the identity of the web site that directed the user to the site (which is state information or as discussed with respect to claim element 1.1, it would have been obvious to use the same technique for other types of information that would constitute state information, Ex. 1002, at ¶ 100). Ex. 1003 at 12:9-16 ("The [page_link.cgi] program call includes two arguments, namely, a destination page identifier (i.e., index) representing the particular page at site 128 to which the user has been directed, and a [referring website code] (i.e., CM1) representing the identity of the web site 122a that directed the user to site 128."); Ex. 1002 at ¶ 236.

(c) ***"wherein said step of embedding is performed by the CGI program"***

Graber teaches this limitation because for each URL on a page which includes the string page_link.cgi," the page_link.cgi program, a CGI program, appends the code of a referring website to the end of the URL. Ex. 1003 at 12:43-48 ("Next, in step 660, for each internal URL in the destination page which includes a string for calling the page_link.cgi program, the page_link.cgi program appends the UNIX symbolic link/CMID code (i.e. CM1) originally passed as an argument to the program to the end of the URL."); Ex. 1002 at ¶ 237.

(9) ***Claim 49***

Claim 49 depends from Claim 47 and, as discussed above, Graber discloses and renders obvious all the elements of Claim 47. Ex. 1002 at ¶ 246. In addition, Claim 49 recites the following:

(a) ***"wherein said converter means further comprises: means for modifying an identified continuation which is an invocation to a CGI program with the identified continuation and the state information passed as arguments, wherein said converter means comprises the CGI program."***

Graber discloses every limitation in this element for the same reasons as in Claim 12. Ex. 1002 at ¶ 247. The functionality of Claim 12 meets the function, and the structure is web server 142 performing the algorithm disclosed at Fig. 6 in boxes 640, 660, and 670. Ex. 1002 at ¶ 247.

2. Ground 2: Graber view of Ibrahim renders claims 3, 16, 29, 42, 53, and 62 obvious

(a) **Overview of Ibrahim**

Ibrahim discloses various solutions to overcome "the fact that the http protocol is stateless, that is, the server does not remember former queries when it processes a new one." Ex. 1004 at 257. One such solution: "The program state and data would be transmitted to the WWW client, both as text to be displayed and, in a more compact form, as a URL hidden in a hyperlink associated with the Step and Continue push-buttons." *Id.* at 258; *id.* at 258, n. 27 ("output the state of the

program in HTML syntax, with step and Continue buttons containing the URL just calculated"). Ibrahim also discloses: "As an alternative to transmitting the whole program data from the server to the client, then back to the server, this information could be stored on the server side, in a temporary file, and have the URL hold a reference to this file." *Id.* at 258; Ex. 1002 at ¶ 250.

(b) **Obvious to combine Graber with Ibrahim**

It would have been obvious to utilize the "alternative" of not transmitting all state information from the server to the client with the disclosure of Graber. Graber and Ibrahim identify the same problem—overcoming statelessness in HTML—and the same solution—passing state information in the URL embedded in hyperlinks of generated HTML pages. Ex. 1003 at 1:45-48; Ex. 1004 at 527-528; Ex. 1002 at ¶ 251.

Ibrahim itself provides specific motivation to modify the transmission of all state information as disclosed in Graber to instead store state information on the server side "and have the URL hold a reference" to the stored information by identifying this "[a]s an alternative to transmitting the whole program data from the server to the client, then back to the server." Ex. 1004 at 258. Thus, Ibrahim

explicitly instructs a POSA to implement storing state information on the server side as an "alternative" to transmitting all information. *Id*; Ex. 1002 at ¶ 252.

(c) **Claims 3, 16, 53, and 62**

Claims 3, 16, 53, and 62 depend from Claims 1, 14, 52, and 60 respectively, and as shown above, Graber discloses and renders obvious every element of Claims 1, 14, 52, and 60. Ex. 1002 at ¶¶ 253-266.

Further each of claims 3, 16, 53, and 62 recite substantially the same language:

(a) ***"storing at least part of the state information in a memory coupled to the server and embedding an index representing said part of the state information in said all identified continuations."***

Graber in view of Ibrahim discloses this claim element. Specifically, as in claim element 1.6, Graber discloses the server transmitting state information, namely the code for a referring web page through the URL. Ex. 1003 at 12:43-53 ("for each internal URL in the destination page which includes a string for calling the page_link.cgi program, the page_link.cgi program appends the [a code for the referring website] originally passed as an argument to the program to the end of the URL. In addition, in step 670 for each external URL in the destination page which includes the "page_link.cgi" string, the page_link.cgi program appends the [a code for the referring website] originally passed to the program followed by a [code]

representing [the current site] to the end of the URL."); Ex. 1002 at ¶ 254. To the extent IBM argues that navigation history is not state information, then for the reasons stated with respect to element 1.1 it would have been obvious for a POSA to use the same technique with other types of information. Ex. 1002, at ¶ 100.

Ibrahim discloses that "[a]s an alternative to transmitting the whole program data from the server to the client, then back to the server, this information could be stored on the server side, in a temporary file, and have the URL hold a reference to this file." Ex. 1004 at 258. Applying this "alternative" method disclosed in Ibrahim to the system of Graber, at least part of the state information, namely the identity of a referring website, would be "stored on the server side, in a temporary file" and "the URL [would] hold a reference to this file." *Id*. That reference is "an index representing said part of the state information." Ex. 1002 at ¶ 255.

Further, with regard to Claims 16 and 62, Graber discloses "program storage device readable by a computer tangibly embodying a program of instructions executable by the computer" for performing the functionality discussed above, namely code running on web server 142 performing the algorithm of Claim 14. Ex. 1002 at ¶¶ 258, 266.

(d) **Claims 29 and 42**

Claim 29 and 42 depend from Claims 28 and 41 respectively, and as shown above, Graber discloses and renders obvious all the limitations of claims 28 and 41. Ex. 1002 at ¶¶ 259-262. Each also recite substantially the same language:

(a) ***"a memory, coupled to the server, for storing at least part of the state information; wherein said converter [logic] / [means] is adapted for embedding an index representing said part of the state information in said all identified continuations"***

Ibrahim discloses this claim element because Ibrahim teaches storing information "on the server side, in a temporary file," (Ex. 1004 at 258), which implicitly requires memory for storing this information. *Id.* Regarding the adaptation of the converter logic/means, the functionality of Claim 3 meets the function, and the structure is web server 142 performing the algorithm of Claim 3. Ex. 1002 at ¶ 260.

3. Ground 3: Graber in view of Ibrahim and Williams renders Claims 4, 5, 17, 18, 30, 31, 43, 44, 63, and 64 obvious.

(a) **Overview of Williams**

Williams discloses the use of Java in e-commerce systems utilizing the World Wide Web, specifically a payment instruction applet at a merchant web site. Ex. 1005 at 12:55-13:42. Williams explains the basics of Sun Microsystem's Java

language, that it "supports programming for the Internet in the form of platform-independent Java applets" providing functionality beyond available with HTML. Ex. 1005 at 10:13-15. Additionally, "[a]pplets execute within a Java-compatible browser (e.g. Netscape Navigator) by copying code from the server to the client." *Id.* at 10:19-21. Moreover, Williams identifies a specific benefit of Java, explaining that Java enables "offloading appropriate processing onto the client for improved performance." *Id.* at 10:2-4; Ex. 1002 at ¶ 267.

(b) **Reasons to Combine Graber/Ibrahim with Williams**

A POSA would have been prompted to utilize and understood the benefits of applying the teachings of Williams regarding Java implementations to the disclosures of Graber and Ibrahim so as to take advantage of Java's off-loading capabilities. As expressly called out by Graber, in considering the implementation of servers (in Graber's case for enrolling users and issuing payments to co-marketers) an architect must ensure scalability, such that the servers handle an increasing number of users. *See* Ex. 1003 at 8:4; Ex. 1002 at ¶ 268. Williams discloses that Java embodiments may improve performance of servers by "offloading appropriate processing onto the client." Ex. 1005 at 10:3-5. Offloading the processing of the server executed services in Graber to the client, such as

page_link.cgi, is consistent with Graber's teachings regarding the importance of creating scalable architecture. See Ex. 1002 at ¶ 268. A POSA would have recognized that offloading server processing of the page_link.cgi program onto the client would have yielded predictable results, namely, improved server performance. *Id*. Thus, the combination of Williams with the disclosures of Graber and Ibrahim is merely combining prior art elements according to known methods to yield predictable results. *Id*. Demonstrating the straight-forward nature of this combination, the '601 Patent provides no details on how to implement a client-side Java program, merely acknowledging that the "Java programming environment is well-known," and stating without elaboration "[a]nother application of downloadable server code to the present invention would be to allow the 'converter 416," disclosed only as a server-side CGI program, "to run on the client." Ex. 1001 at 16:17-18, 16:30-33; Ex. 1002 at ¶ 268.

(c) **Claims 4, 17, 54, and 63**

Claims 4, 17, 54, and 63 depend from Claims 1, 14, 51, and 60 respectively, and as shown above, Graber discloses and renders obvious every element of Claims 1, 14, 51, and 60. Ex. 1002 at ¶¶ 269-283. Further each of claims 4, 17, 51, and 63 recite substantially the same language:

(a) ***"further including the step of "dynamically downloading computer program code to the client to perform the step of embedding, which is responsive to said step of communicating the output to the client"***

As discussed above, Graber discloses the method of Claim 1 including embedding the state information in all identified continuations. *See supra* at Element 1.6; Ex. 1003 at FIG. 6, 13:45-55. To the extent IBM argues that navigation history is not state information, as discussed with respect to element 1.1, a POSA would find it obvious to use the same technique with other types of information. Ex. 1002, at ¶ 100. Williams provides further details for how the system may embed all identified continuations with state information. Namely, Williams describes the execution of Java applets, specifically noting that they execute on a "Java-compatible browser (e.g. Netscape Navigator) by copying code form the server to client." Ex. 1004 at 10:20-21. This teaching is consistent with the observation of the '601 Patent that the use of the Java programming environment including downloadable server programs called applets was well known at the time of the invention. Ex. 1004 at 16:14-18.

Applying this teaching to Graber, which discloses that the applicable server must be able to handle a large number of user stations (Ex. 1003 at 7:4), Williams discloses dynamically downloading computer code to the client to perform the step

of embedding, thus reducing the server load and improving performance. Ex. 1005 at 10:19-21 ("Applets execute within a Java-compatible browser (e.g. Netscape Navigator) by copying code from the server to the client."). The step would follow, and thus be responsive to, communicating the output to the client. Ex. 1002 at 272.

Further, with regard to Claims 17 and 63, Graber discloses a "program storage device readable by a computer tangibly embodying a program of instructions executable by the computer" for performing the functionality discussed above, namely code running on web server 142 performing the algorithm of Claim 14. *Id.* at ¶¶ 274, 283.

(d) **Claim 30 and 43**

Claim 30 and 43 depend from Claims 27 and 40 respectively, and as discussed above, Graber discloses and renders obvious all the limitations of Claims 27 and 40. Claims 27 and 40 also recite substantially the same language:

(a) ***"wherein said communication [logic] / [means] communicates the output without embedded state information from the server to the client; and wherein the server is adapted for dynamically downloading said converter [logic] / [means] to the client for execution."***

Graber in view of Ibrahim and Williams renders this additional limitation obvious. As discussed above with regard to Claim 4, applying the disclosure of Williamson to Graber in view of Ibrahim results in a system in which the server

side code for performing the embedding step is adapted to a Java applet downloadable by the server to the client. Ex. 1002. at ¶ 276. In such a case, the web page containing the converter logic/means when communicated to the client does not include embedded information. *Id.* Therefore, the communication logic/means, namely the server performing the function discussed above, would communicate the output (i.e., HTML file) from the server to the client without the embedded state information. *Id.*

(e) **Claim 5, 18, 55, 64**

Claims 5, 18, 55, and 64 depend from Claims 1, 14, 54, and 60 respectively, and as shown above, Graber discloses and renders obvious every element of Claims 1, 14, and 60 and Graber in view of Ibrahim and Williams discloses and renders obvious every element of Claim 54. Further, Claims 5, 18, 55, and 64 also recite substantially the same language:

(a) ***"storing at least part of the state information in a memory coupled to the client and wherein said step of embedding includes embedding an index representing said part of the state information"***

Graber in view of Ibrahim and Williams discloses the limitations of this claim element. Specifically, as in claim 3 above, applying the "alternative" method disclosed in Ibrahim to the system of Graber, at least part of the state information, namely the code for a referring website, would be "stored on the server side, in a

temporary file" and "the URL [would] hold a reference to this file." *See supra*, section V.D.2.(c); *See* Ex. 1002 at ¶ 285. That reference is "an index representing said part of the state information." As in Claim 4 above, applying the disclosure of Williams to this system would result in shifting the embedding functionality from the server to the client side. *See* supra V.D.3.(c). This shift would result in shifting the memory associated with the embedding functionality, resulting in storage of state information in a memory coupled to the client. *Id*; Ex. 1002 at ¶ 285.

Further, with regard to Claims 18 and 64, Graber discloses "program storage device readable by a computer tangibly embodying a program of instructions executable by the computer" for performing the functionality discussed above, namely code running on web server 142 performing the algorithm discussed with regard to Claim 14. Ex. 1002 at ¶¶ 288, 297.

(f) **Claims 31 and 44**

Claims 31 and 44 depend from Claims 30 and 43 respectively, and as discussed above, Claims 30 and 43 are rendered obvious by Graber in view of Ibrahim and Williams. Ex. 1002 at ¶¶ 290-292. Claims 31 and 41 also contain substantially similar language:

(a) ***a memory, coupled to the client, for storing at least part of the state information; wherein said converter [logic] / [means] is adapted for embedding an index representing said part of the state information in said all identified continuations."***

Graber in view of Ibrahim and Williams renders this claim obvious for at least the reasons discussed above with regard to Claim 5. Ex. 1002 at ¶ 290. Applying the disclosure of Williamson to this system would result in shifting the embedding functionality from the server to the client side, including shifting the memory associated with the embedding functionality, resulting in storage of state information in a memory couple to the client. *Id.* Regarding the adaptation of the converter logic/means, the functionality of Claim 5 meets the function, and the structure is the web server 142 in Graber performing the algorithm discussed with regard to Claim 5. *Id.*

VI. CONCLUSION

For at least the reasons set forth above, claims 1, 2-8, 11, 12, 14-21, 24, 25, 27-34, 37, 38, 40-45, 47-49, 51-57, and 60-66, of the '601 patent are unpatentable under 35 U.S.C. §§ 102(a) and/or 103(a). Petitioner therefore respectfully requests that the Board institute the requested *Inter Partes* review of the '601 Patent.

Dated: February 10, 2016

Respectfully submitted,

/Richard S. Zembek/

Richard S. Zembek (Reg. No. 43,306)
NORTON ROSE FULBRIGHT US LLP
1301 McKinney, Suite 5100
Houston, TX 77010
Tel: 713.651.5151 Fax: 713.651.5246
richard.zembek@nortonrosefulbright.com
Attorneys for Petitioner
KAYAK Software Corp., OpenTable, Inc., Priceline.com LLC, and The Priceline Group Inc.

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6(e) and 37 C.F.R. § 42.105(a), the undersigned certifies that on February 10, 2016, a complete copy of this Petition for *Inter Partes* Review and all exhibits were served on Patent Owner at the correspondence addresses of record listed below by EXPRESS MAIL®.

c/o Kevin M. Jordan
Intellectual Property Law Dept.
IBM Corp.
P O BOX 218
Yorktown Heights, NY, 10598

Additionally, the undersigned certifies that on February 10, 2016, a complete copy of this Petition for *Inter Partes* Review and all exhibits were served on Patent Owner's below-listed counsel of record in the Litigation at the address listed below by Federal Express®:

Richard L. Horwitz
Potter Anderson & Corroon, LLP
1313 N. Market St., Hercules Plaza, 6th Flr.
P.O. Box 951
Wilmington, DE 19899-0951

/Richard S. Zembek/

Richard S. Zembek (Reg. No. 43,306)

Attorney for Petitioner